

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) A method of distributing portions of a query over two or more execution engines, the method comprising:

receiving an input query into a first analysis engine of serially cascaded analysis engines;

identifying with ~~a~~ the first analysis engine, a portion of the input query that can be processed by a first execution engine;

compiling the identified portion of the input query forming a first compiled portion;

rewriting the input query to form a first rewritten query wherein the identified portion of the input query is removed from the input query and replaced by a first placeholder;

passing an entire portion of the first rewritten query including the first placeholder to a second analysis engine of the serially cascaded analysis engines;

identifying with the second analysis engine, a portion of the first rewritten query that can be processed by a second execution engine; and

compiling the identified portion of the first rewritten query generating a second compiled portion wherein the input query is distributed over the first execution engine and the second execution engine for sequential execution.

2. (Original) The method of claim 1, further comprising:

rewriting the first rewritten query to form a second rewritten query wherein the identified portion of the first rewritten query is removed from the first rewritten query and replaced by a second placeholder.

3. (Currently Amended) The method of claim 1, wherein the rewriting act further comprises wrapping the second compiled portion into the first compiled portion of the input query to produce a nested query for sequential execution.

4. (Original) The method of claim 1, wherein the first compiled portion and the second compiled portion may be executed over different data sources.

5. (Original) The method of claim 1, wherein a first analysis engine is a structured query language based engine and a second analysis engine is an extensible markup language based engine.

6. (Original) The method of claim 1, further comprising:

executing partially the second compiled portion using the second execution engine forming the combination of second interim results and the first placeholder, wherein the first placeholder remains an unexecuted part of the second compiled portion;

generating a call from the second execution engine to the first execution engine requesting the data corresponding to the first placeholder;

executing the first compiled portion using the first execution engine to form first interim results corresponding to the first placeholder and satisfying the call;

providing the first interim results to the second execution engine; and

substituting the first interim results for the first placeholder forming the combination of second interim results and first interim results comprising combined input query results.

7. (Original) The method of claim 6, wherein the first execution engine and the second execution engine operate on queries comprising different data models.

8. (Original) The method of claim 1, further comprising:

executing partially the first compiled portion using the first execution engine forming first interim results;

generating a call from the first execution engine to the second execution engine requesting the data corresponding to an unidentified portion of the input query;

executing the second compiled portion using the second execution engine to form second interim results and satisfying the call;

providing the second interim results to the first execution engine; and

combining the first interim results with the second interim results to form combined input query results.

9. (Original) The method of claim 8, wherein the first execution engine and the second execution engine operate on queries comprising different data models.

10. (Currently Amended) A system for distributive processing of an input query, the system comprising:

two or more serially cascaded analysis engines for separating out portions of the input query that can be compiled and executed;

two or more serially cascaded execution engines for operation on the input query; and

access to one or more data sources, wherein the two or more analysis engines operate to independently identify and compile one or more portions of the input query wherein:

at least one of the two or more analysis engines rewrites the input query to remove the portion of the input query that corresponds to an execution engine and replaces the portion with a placeholder, and

the rewritten query, including the placeholder is input to a subsequent analysis engine in the two or more serially cascaded analysis engines; and

the two or more execution engines sequentially process the one or more compiled portions of the input query such that partial query results from one execution engine are passed to a subsequent execution engine and combined to form overall input query results.

11. (Currently Amended) A computer-readable media containing instructions, which when run on a computer, execute a method of distributing portions of a query over two or more execution engines, the method comprising:

receiving an input query; into a first analysis engine of a serial cascade of analysis engines;

identifying with ~~a~~ the first analysis engine, a portion of the input query that can be processed by a first execution engine;

compiling the identified portion of the input query forming a first compiled portion;

rewriting the input query to form a first rewritten query wherein the identified portion of the input query is removed from the input query and replaced by a first placeholder;

passing an entire portion of the first rewritten query including the first placeholder to a second analysis engine of the serial cascade of analysis engines;

identifying with the second analysis engine, a portion of the first rewritten query that can be processed by a second execution engine; and

compiling the identified portion of the first rewritten query generating a second compiled portion wherein the input query is distributed over the first execution engine and the second execution engine; and

wherein the rewriting act further comprises wrapping the second compiled portion into the first compiled portion to produce a nested query for sequential execution.

12. (Original) The computer-readable medium of claim 11, further comprising:

executing partially the second compiled portion using the second execution engine forming the combination of second interim results and the first placeholder, wherein the first placeholder remains an unexecuted part of the second compiled portion;

generating a call from the second execution engine to the first execution engine requesting the data corresponding to the first placeholder;

executing the first compiled portion using the first execution engine to form first interim results corresponding to the first placeholder and satisfying the call;

providing the first interim results to the second execution engine; and

substituting the first interim results for the first placeholder forming the combination of second interim results and first interim results comprising combined input query results.

13. (Original) The method of claim 12, wherein the first execution engine and the second execution engine operate on queries comprising different data models.

14. (Original) The method of claim 12, wherein the first compiled portion and the second compiled portion are executed over different data sources.

15. (Original) The computer readable medium of claim 11, further comprising:

executing partially the first compiled portion using the first execution engine forming first interim results;

generating a call from the first execution engine to the second execution engine requesting the data corresponding to an unidentified portion of the input query;

executing the second compiled portion using the second execution engine to form second interim results and satisfying the call;

providing the second interim results to the first execution engine; and

combining the first interim results with the second interim results to form combined input query results.

16. (Original) The method of claim 15, wherein a first analysis engine is a structured query language based engine and a second analysis engine is an extensible markup language based engine.